WHAT IS CLAIMED IS

1. A managing method for an re-order buffer in an out-of-order execution processor for predicting the flow of a program by a branch prediction, finding out a next executable instruction from a instruction string in the program and speculatively executing the instruction on the basis of dependence relationship between the prediction and the instruction, in which said re-order buffer rewrites an execution result according to a program order and the end of the instruction is notified from each of function units containing a branching unit and a load unit to said re-order buffer by using WRB number corresponding to an entry number of said re-order buffer, which comprises the steps of:

managing the latest speculation state of a load instruction issued to said load unit by said load unit on the basis of a branch prediction success/failure signal output from said branching unit and suppressing notification to said re-order buffer by said load unit, as to a subsequent load instruction of a branch instruction for which the branching prediction has failed, on the basis of the WRB number of the subsequent load instruction even when the processing of the load instruction concerned is finished, and

re-using an entry stored with the subsequent instruction of the branching-prediction failed branching instruction by the re-order to store a new instruction before the end notification based on the WRB number of the entry concerned is received.

2. The managing method as claim in claim 1, wherein a control signal for discriminating non-speculative execution/speculative execution every instruction, which corresponds to a branching level, is generated in a

instruction fetch/decode unit, the branching level being set to zero when the instruction concerned is a instruction for non-speculative execution, and the branching level being set to a value of being 1 or more which is determined by the number of branching instructions interposed between the instruction for the non-speculative execution and the instruction for the speculative execution when the instruction concerned is a instruction for speculative execution, the control signal thus generated is held in said re-order buffer and said load unit, and the branching level is decremented by 1 in said re-order buffer and said load unit every time when a branch-prediction failure signal is output from said branching unit, thereby managing the latest instruction speculation state.

3. The managing method as claimed in claim 2, wherein said load unit has plural entries each of which holds a load instruction issued to said load unit together with the branching level and the WRB number thereof, the entries having the branching level of 1 or more set to a cancel state when a branch-prediction failure signal is output from said branching unit, and with respect to load instructions held in the entries under the cancel state, a notification to the re-order on the basis of the WRB numbers of the load instructions concerned is suppressed even when the processing of the load instructions is finished.

4. An out-of-order execution processor for predicting the flow of a program by a branch prediction, finding out a next executable instruction from a instruction string in the program and speculatively executing the instruction on the basis of dependence relationship between the prediction and the instruction, in which a re-order buffer in said processor rewrites an execution

result according to a program order and the end of the instruction is notified from each of function units containing a branching unit and a load unit to said re-order buffer by using WRB number corresponding to an entry number of said re-order buffer, which comprises:

managing means for managing the latest speculation state of a load instruction issued to said load unit on the basis of a branch prediction success/failure signal output from said branching unit and suppressing notification to said re-order buffer, as to a subsequent load instruction of a branch instruction for which the branching prediction has failed, on the basis of the WRB number of the subsequent load instruction even when the processing of the load instruction concerned is finished, said managing means being contained in said load unit, wherein the re-order re-uses an entry stored with the subsequent instruction of the branching-prediction failed branching instruction to store a new instruction before the end notification based on the WRB number of the entry concerned is received.

5. The processor as claim in claim 4, wherein a control signal for discriminating non-speculative execution/speculative execution every instruction, which corresponds to a branching level, is generated in a instruction fetch/decode unit, the branching level being set to zero when the instruction concerned is a instruction for non-speculative execution, and the branching level being set to a value of being 1 or more which is determined by the number of branching instructions interposed between the instruction for the non-speculative execution and the instruction for the speculative execution when the instruction concerned is a instruction for speculative execution, the control signal thus generated is held in said re-order buffer

and said managing means, and the branching level is decremented by 1 in said re-order buffer and said managing means every time when a branch-prediction failure signal is output from said branching unit, thereby managing the latest instruction speculation state.

6. The processor as claimed in claim 5, wherein said managing means has plural entries each of which holds a load instruction issued to said load unit together with the branching level and the WRB number thereof, the entries having the branching level of 1 or more set to a cancel state when a branch-prediction failure signal is output from said branching unit, and with respect to load instructions held in the entries under the cancel state, a notification to the re-order on the basis of the WRB numbers of the load instructions concerned is suppressed even when the processing of the load instructions is finished.